Rethinking Stateful Stream Processing with RDMA

Bonaventura Del Monte[1], Steffen Zeuch[1,2], Tilmann Rabl[3], Volker Markl[1,2]

[1]Technische Universität Berlin · [2]DFKI GmbH · [3]Hasso-Platner Institute · [3]Potsdam Universität

bdelmonte@tu-berlin.de · steffen.zeuch@dfki.de · tilmann.rabl@hpi.de · volker.markl@tu-berlin.de

Overview and Motivation

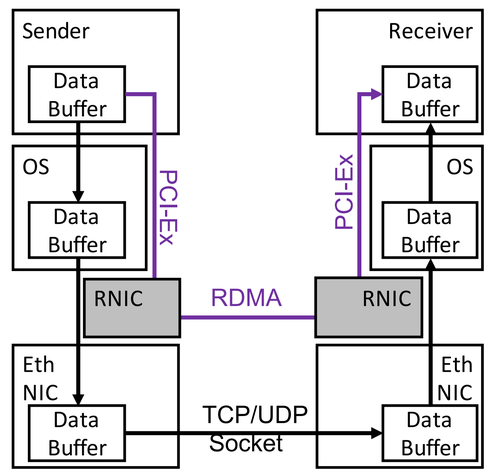
High-speed networks and RDMA have invalidated the common assumption that network is often the bottleneck for scale-out SPEs.

Current SPEs design is *RDMA-unfriendly*, as it relies on costly data re-partitioning to scale-out.

We propose Slash: an SPE suited for native RDMA acceleration that scales out by omitting the expensive data re-partitioning demands

What is Remote Direct Memory Access? How does it help?

- OS kernel stack bypass and zero-copy transfer.
- Message-oriented via one-sided and two-sided verbs API.
- Current DMS use RDMA to accelerate batch OLAP and OLTP.

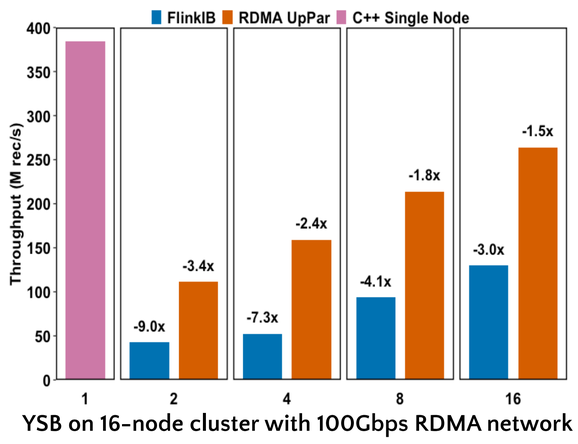


Why cant SPE benefit from RDMA acceleration?

Data repartitioning is costly.

Swapping socket-based with RDMA communication does not make SPEs faster.

Poor data and code locality induced by message-passing.



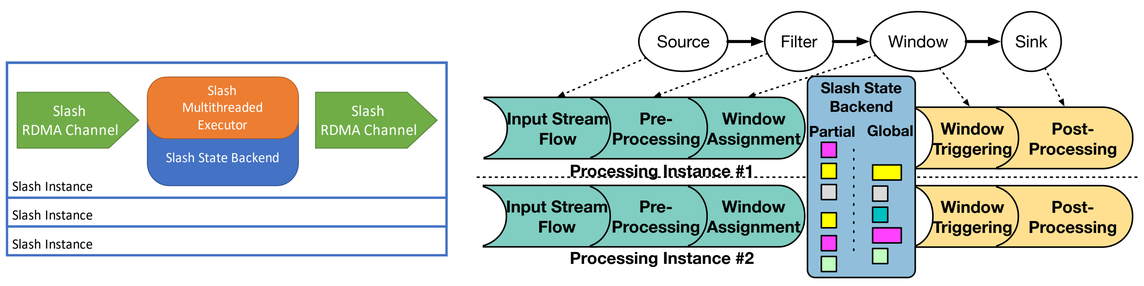
YSB on 16-node cluster with 100Gbps RDMA network

Design Challenges and Solutions

- Efficient streaming computations: Replace data re-partitioning with RDMA-enabled late merge.
- Efficient data transfer: RDMA depends on low-level factors.
- Consistent stateful computations: Progress tracking and exactly-once state updates.

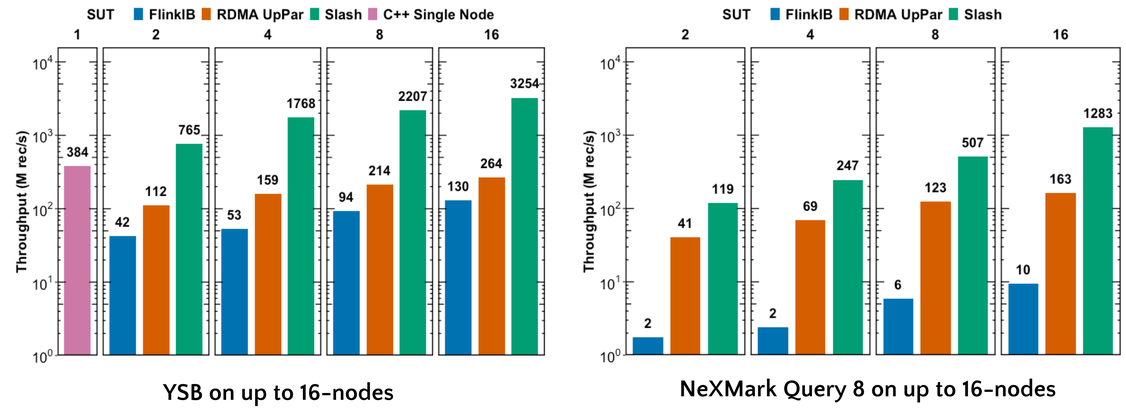
Slash: our RDMA-enabled SPE

Guiding Design Principle: make the common case fast!



Eager computation of partial state and lazy merging to obtain a consistent state.

Performance evaluation



YSB on up to 16-nodes

NeXMark Query 8 on up to 16-nodes

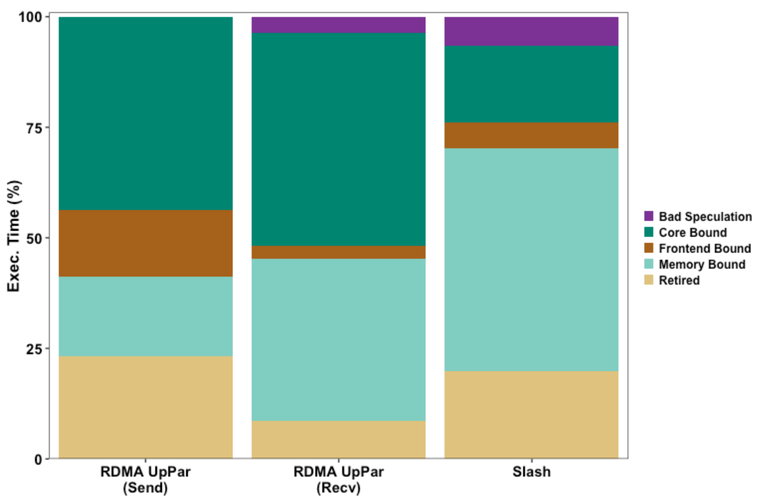
Slash outperform baseline solutions on common streaming workloads.

Performance gain explained

	IPC	Instr./ Rec.	Cyc./ Rec.
RDMA	0.6	166	274
UpPar	0.4	78	276
Slash	0.9	42	53

RDMA Up-Par is bound by partitioning and network speed.

Slash is bound by memory performance.



Take Home

- We provide a new system design for RDMA-accelerated stateful stream processing.
- We apply RDMA native acceleration by redesigning internal data structures to avoid data repartitioning and use lazy merge.
- We show that an up to a factor of 25 increase in throughput compared to the strongest baseline.
- We perform a drill-down analysis to explain why our solutions performs better than our strongest baseline.

Preprint is available!

Slash is part of NebulaStream: our Data Management Platform for the IoT. Check out the preprint of our paper as well as the NebulaStream project!

https://nebula.stream

Follow us on Twitter @nebulastream

NebulaStream

SCAN ME

Acknowledgements

This work was funded by the German Ministry for Education and Research as BIFOLD – Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and 01IS18037A) and by the German Research Foundation as FONDA (ref. 414984028).

German Research Center for Artificial Intelligence

Hasso Plattner Institut

Digital Engineering · Universität Potsdam

BIFOLD

The Berlin Institute for the Foundations of Learning and Data

Deutsche Forschungsgemeinschaft

Bundesministerium für Bildung und Forschung